

Navigating the Semantic Horizon using Relative Neighborhood Graphs

Amaru Cuba Gyllensten and Magnus Sahlgren

Gavagai

Bondegatan 21

116 33 Stockholm

Sweden

{amaru|mange}@gavagai.se

Abstract

This paper is concerned with nearest neighbor search in distributional semantic models. A normal nearest neighbor search only returns a ranked list of neighbors, with no information about the structure or topology of the local neighborhood. This is a potentially serious shortcoming of the mode of querying a distributional semantic model, since a ranked list of neighbors may conflate several different senses. We argue that the topology of neighborhoods in semantic space provides important information about the different senses of terms, and that such topological structures can be used for word-sense induction. We also argue that the topology of the neighborhoods in semantic space can be used to determine the *semantic horizon* of a point, which we define as the set of neighbors that have a direct connection to the point. We introduce *relative neighborhood graphs* as method to uncover the topological properties of neighborhoods in semantic models. We also provide examples of relative neighborhood graphs for three well-known semantic models; the PMI model, the GloVe model, and the skipgram model.

1 Introduction

Nearest neighbor search is fundamental operation in data mining, in which we are interested in finding the closest points (to some given reference point). Formally, if we have a reference point r and a set of other points P in a metric space M with some distance function d (or similarity function s), the nearest neighbor search task is to find the point $p \in P$ that minimizes $d(p, r)$. In k -nearest neighbor search (k -NN), we want to find the k closest points to some given reference point. Nearest neighbor search is a well-studied task, and in particular the complexity of the task (a linear search has a running time of $\mathcal{O}(Ni)$ where N is the cardinality of P and i the complexity of the distance function d) has generated a lot of research (Bentley, 1975; Arya et al., 1998; Indyk and Motwani, 1998).

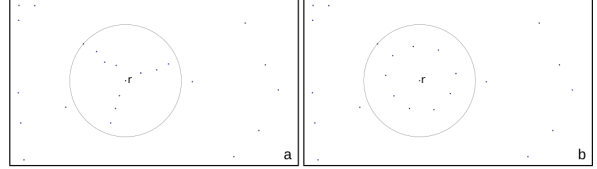


Figure 1: Examples of neighborhoods with a clear branching structure (a) and without (b).

The problem we are concerned with in this paper is not the complexity of nearest neighbor search, but the question *how to identify the internal structure of neighborhoods defined by the nearest neighbors*. The problem with a normal k nearest neighbor search is that the result (a sorted list of the k nearest neighbors) does not say anything about the internal structure of the neighborhood. Consider spaces a and b in Figure 1. A nearest neighbor search for the reference point r in these two spaces will generate the exact same result, despite the fact that the neighborhoods are very different with regards to their internal structure (the neighbors in space a display a distinct branching structure, whereas the neighbors in space b are distributed evenly across the space). Such structural properties of nearest neighborhoods can be very important.

We propose to use *relative neighborhood graphs* in order to identify the structural properties of nearest neighborhoods. The use of relative neighborhood graphs also provides a partial solution to the problem of finding a relevant k for a given reference point. Again, consider the neighborhoods in Figure 1. The choice of $k = 10$ in these spaces is completely arbitrary, and could be argued to be erroneous, since there are in fact 12 relevant neighbors in both spaces. Another way to approach the nearest neighbor search task is to use a radius around the reference point, so that only points within that radius are considered to be neighbors. However, setting a global threshold t seems just as arbitrary as setting a global number of neighbors k . Ideally, t or k should be determined based on the structural properties of the nearest neighborhood around the reference point. We refer to this

factor as the *horizon* with respect to the reference point.

Although a relative neighborhood graph is a general method for defining and structuring nearest neighborhoods, we are in this paper primarily interested in its application to nearest neighbor searches in *distributional semantic models* that collect and represent co-occurrence statistics in high-dimensional vector spaces. The main operation in such models is nearest neighbor search, which is used for finding terms that have similar co-occurrence behavior. However, a ranked list of neighbors does not provide any information on whether the neighbors belong to several different senses. This problem has been misinterpreted as a shortcoming of the distributional representation (Erk and Padó, 2010). However, as we will demonstrate in this paper, this is not a shortcoming of the distributional representation, but of the *mode of querying* the distributional model. We argue that information about the different usages (i.e. senses) of a term is encoded in the structural properties of the nearest neighborhoods, and that a relative neighborhood graph is a viable tool for uncovering such structural properties.

2 Distributional Semantics and Nearest Neighbor Search

Collecting and comparing co-occurrence statistics for terms in language has become a standard approach for computational semantics, and is now commonly referred to as *distributional semantics*. There are many different types of models that can be used for this purpose, but their common objective is to represent terms as vectors that record (some function of) their distributional properties. The standard approach for generating such vectors is to collect distributional statistics in a *co-occurrence matrix* that records co-occurrence counts between terms and contexts. The co-occurrence matrix is then subject to various types of transformations, ranging from the application of simple frequency filters or association measures like pointwise mutual information, to matrix factorization or regression models. The resulting representations are referred to as *distributional vectors*, and are typically dense with a dimensionality that is considerably lower than that of the original co-occurrence matrix.

The distributional vectors are used to compute similarity between terms. There are many ways to compute similarity or distance between points in vector space; the cosine of the angle between vectors is often the preferred metric in distributional semantics because of its simplicity and because it normalizes for vector length. Computing the similarity between distributional vectors using the cosine measure gives us a score ranging from -1 —

negatively collinear — to 1 — positively collinear — taking the value 0 if the vectors are orthogonal.

We can thus use a distributional semantic model to quantify the similarity between any given terms. If the set of given terms is the *entire* set of terms in our model, we are in effect performing a nearest neighbor search. This is a particularly important operation in distributional semantics, since it answers the question "which other terms are similar to this one?", and this is a central question in semantics; lexica and thesauri are built with the main purpose of answering this question, and a nearest neighbor search in a distributional semantic model could therefore be seen as a compilation step in a distributional lexicon.

The result of a nearest neighbor search in a distributional semantic model is often presented as a list of (the top k) neighbors, sorted by descending similarity with the target term. Table 1 illustrates typical sorted nearest neighbor lists produced with three different kinds of distributional semantic models: a vanilla-flavored model based on (positive) Pointwise Mutual Information (PMI),¹ the skip-gram model (Mikolov et al., 2013), and GloVe (Pennington et al., 2014).

Table 1: Sorted list of the nearest neighbors to "suit" in different distributional models. Different fonts represent different meanings of "suit."

PMI	GloVe	skipgram
suits	suits	suits
dress	lawsuit	lawsuit
jacket	filed	countersuit
wearing	case	classaction
hat	wearing	doublebreasted
trousers	laiming	skintight
costume	lawsuits	necktie
shirt	alleging	wetsuit
pants	alleges	crossbone
lawsuit	classaction	lawsuits

In the vanilla-flavored model, the distributional vector of a word is given by its (positive) PMI with regards to all other words that have occurred within a context window of 2 words to the left and 2 words to the right. That is, a vector for a word a corresponds to the information an observation of a gives when predicting surrounding words. *Positive* PMI means that negative values are discarded, and only positive PMI values are retained. The cosine similarity of two distributional vectors thus gives a measure of how similar the information gained by observing the corresponding words are. As a

¹For observations a and b , $\text{pmi}(a,b) = \log \frac{p(a,b)}{p(a)p(b)}$. The probabilities are often replaced in distributional semantic models by co-occurrence counts of a and b and their respective frequency counts.

way to speed up later computations we apply a Gaussian random projection to reduce the dimensionality down to 2000.

GloVe on the other hand tries to find distributional vectors such that their dot product approximates their log probability of co-occurring is motivated by the fact that the logarithm of ratios equals the difference of logarithms, which makes the vector differences meaningful in that they encode (logarithms of) ratios of probabilities. Reframed as a weighted least squares problem, where rare co-occurrences are weighted down, it can be solved by standard methods. The performance is comparable to the skip-gram model, and it performs particularly well on word-analogy tasks (Pennington et al., 2014).

The objective of the skipgram model is to maximize the probability of observing all context-word pairs given that the probability of one observation of a word c in the context of t is given by $\frac{\exp(w_c^T v_t)}{\sum_{i \in V} \exp(w_i^T v_t)}$ where v_a and u_a denotes the "input" and "output" vectors of the word a , and V is the vocabulary. The embeddings are found using stochastic gradient descent and hierarchical softmax combined with negative sampling and subsampling. Exactly how these methods compose is still unclear, and puts into question what the underlying model actually is (Levy and Goldberg, 2014). Regardless, the skip-gram model delivers state of the art performance on a multitude of tasks, with very low-dimensional vectors (Baroni et al., 2014).

Table 1 lists the 10 nearest neighbors to *suit* in three different distributional semantic models using the entire Wikipedia as data.² As can be expected, there are both similarities and dissimilarities between these neighborhoods; "suits" and "lawsuit" occur among the 10 nearest neighbors to "suit" in all three models, whereas other terms are specific for one particular model. Yet all three models feature neighbors of "suit" that represent different senses: the way "suit" is not related to "jacket" in the same way it is related to "lawsuit".

It has been argued that distributional semantic models that represent terms by a single vector cannot adequately handle polysemy, since they conflate several different usage patterns in one and the same vector (Erk and Padó, 2010; Véronis, 2004). Examples like the one above is often cited as evidence. We argue that this critique is unfounded and misinformed, and that it is *the mode of querying* the distributional semantic model that can be susceptible to problems with polysemy. As the above example demonstrates, querying distributional semantic models by k -NN conflates differ-

ent usages of terms. The reason for this seems quite obvious: simply ranking the nearest neighbors by similarity (or distance) ignores any local structures of the neighborhood. If "suit" has as neighbors both "dress" and "lawsuit", which represent two distinct types of usages of "suit", there will be a *structural* distinction in the neighborhood of "suit" between these different neighbors, since they will be mutually unrelated (i.e. there is a similarity between "suit" and "dress" and between "suit" and "lawsuit", but *not* between "dress" and "lawsuit").

k -NN also gives rise to another problem related to polysemy in distributional semantic models. The problem is that the most frequent senses will populate the top of the nearest neighbor list, while the less frequent senses will not appear until further down the list, and if we set a too restrictive k , we will only see neighbors relating to the most frequent sense. Consider, for example, a term such as "suit", which, as we have seen above, may appear in (at least) two different senses: in usages related to *law* and in usages related to *clothes* (or *garment*). The distributional vector can be thought of as a sum $v_{suit} = f_{suit|law}v_{suit|law} + f_{suit|clothes}v_{suit|clothes}$, where $v_{suit|law}$ is an idealized notion of the *true* distributional vector of "suit" in the *law*-sense, and $f_{suit|law}$ is the relative frequency of this sense.³ From there one can easily argue that a similarity such as $s(v_{suit}, v_{garment})$ is actually a weighted composite of the similarities $s(v_{suit|law}, v_{garment})$ and $s(v_{suit|clothes}, v_{garment})$.⁴ If "suit" occurs predominantly in the *law*-sense in our corpus, the k -NN neighborhood of "suit" will be dominated by words pertaining to its *law*-sense, while the less frequent senses might not be present at all. A misguided k may thus obscure any other, less frequent, senses of a term.

Another problem with setting a global k in distributional semantic models is that some terms will have a much denser neighborhood than others. Using the same k for all terms therefore seems ill-advised; terms with a dense neighborhood warrant a larger k than those with a sparse neighborhood. As we have already touched upon in the introduction, determining k is a fundamental question in k -NN, for which there seems to be no clear solution. A more informed approach compared to setting a global k would be to consider the distribution of distances/similarities and attempt to find a gap in the distribution at which to cut off the list. However, the distribution of similarities in distributional semantic models typically does not

²We use a Wikipedia dump from 2010 as data in this and following experiments.

³Weighting schemes muddles this notion quite a bit, but we think the general intuition still holds.

⁴In the case of cosine similarity this follows nicely from the distributive property of dot products: $v = av_1 + bv_2$, $s(v, w) = \frac{v \cdot w}{\|v\| \|w\|} = \frac{a(v_1 \cdot w) + b(v_2 \cdot w)}{\|v\| \|w\|}$

have any clear gaps, as exemplified in Figure 2.

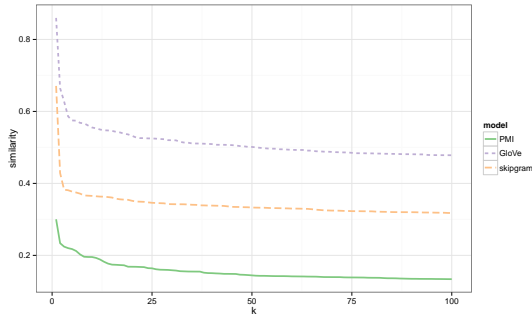


Figure 2: Distribution of similarities for the 100 nearest neighbors of the word *suit* in the three distributional semantic models used in this paper.

Note that the curves behave approximately the same in all three models; there are a few (one or two) very close neighbors, and then the similarities decrease very slowly. The difference in magnitude of the similarities between the models is not peculiar for the word “suit”. On the contrary, PMI, GloVe, and the skipgram model produce vector spaces with different inherent densities. Figure 3 shows both the similarities to 1000 randomly selected points, as well as the similarities to the 10 nearest neighbors to 1000 randomly selected points. The skipgram model produces the highest similarity scores both for related and unrelated points, while the PMI model produces the lowest scores for both related and unrelated points. The GloVe model is in between. All models show a more or less clear distinction between the average similarities to randomly chosen points and the average similarities to the nearest neighbors. This distinction suggests that it might be possible to use the expected similarity to a randomly selected point as a cut-off threshold for k -NN. However, such a global estimate will not be suitable for all terms, for the very same reason alluded to above; different terms have different densities of their neighborhoods. Furthermore, it seems as if the PMI model distinguishes more clearly between the related and the unrelated points, with the skipgram model having the most outliers. This suggests a global estimate might be more useful in some types of models (like the standard PMI model) than in others (like the skipgram model).

3 Word-sense Induction

Selecting a relevant k for a given term and grouping the neighbors according to which senses they represent is an example of *word-sense induction*. Distributional semantic models are well suited for this task, and there have been a number of different approaches suggested in the literature, which can roughly be divided into *context clustering* and *word*

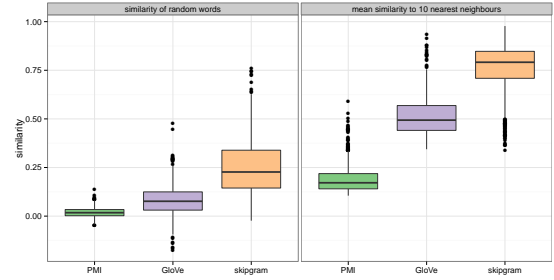


Figure 3: Boxplots of the similarities of 1000 randomly picked word pairs (right), and of the mean similarities to the 10 nearest neighbors for 1000 randomly chosen words (left) in the three distributional semantic models used in this paper.

clustering approaches. Context clustering does not operate on nearest neighbor lists, but instead clusters representations of each individual occurrence of a term. (Schütze, 1998) is one of the earliest examples of a context clustering approach, in which *context vectors* (the centroid of the distributional vectors of the terms that occur in the context) for a given term are clustered into a set of *sense vectors* that represent the induced senses. Other examples of context clustering include (Purandare and Pedersen, 2004), (Velldal, 2005), (Reisinger and Mooney, 2010), (Pedersen, 2010), and (Jurgens and Stevens, 2010).

In contrast to context clustering, word clustering clusters the nearest neighbors into sense groups, and are thus the type of approach that is most relevant for our purposes. The earliest example of a word clustering approach is *distributional clustering* (Pereira et al., 1993), which clusters nouns that occur as heads of direct objects of verbs according to their distributional similarity. The resulting noun clusters for a verb can be interpreted as a representation of the different senses of the verb.

Another example of a word clustering approach is *clustering by committee* (Pantel and Lin, 2002), which is a distributional clustering procedure in several steps. The first step is to use average-link clustering to recursively cluster the nearest neighbors into a set of clusters called *committees*. The committees are then used to define clusters by iteratively adding committees whose similarity to the term exceeds a certain threshold, and that is not too similar to any other added committee. For each added committee, its features are also removed from the distributional representation of the lexeme. This last step ensures that the clusters do not become too similar, and that clusters representing less frequent senses can be discovered.

The idea of iteratively removing features from the distributional vector when a sense cluster has been formed is also present in (Dorow and

Widdows, 2003), who use a graph-based clustering method (Markov clustering (van Dongen, 2000)) to cluster the nearest distributional neighbors of a lexeme. Another graph-based approach to word-sense induction is the *HyperLex* algorithm (Véronis, 2004), which constructs a graph connecting all pairs of terms that co-occur in the context of an ambiguous term. The resulting graph contains highly connected components (hubs), which represent the different senses of the term. (Agirre et al., 2006) compares HyperLex to *PageRank* (Brin and Page, 1998) and demonstrates that the two methods perform similarly on a word-sense induction task. Other examples of graph-based approaches include (Biemann, 2006), (Klapaftis and Manandhar, 2008), and (Marco and Navigli, 2013).

There have also been several attempts to use various types of matrix factorization to perform word sense induction. The idea is that the factorization uncovers a set of global senses in the form of the latent factors, and that the sense distribution for a given term can be described as a distribution over these latent factors. (Brody and Lapata, 2009), (Séaghdha and Korhonen, 2011), (Yao and Van Durme, 2011), and (Lau et al., 2012) use different versions of *Latent Dirichlet Allocation* to produce the factorization, while (Dinu and Lapata, 2010) and (Van de Cruys and Apidianaki, 2011) instead experiment with *non-negative matrix factorization*.

(Tomuro et al., 2007) argues that clustering approaches like distributional clustering or clustering by committee may produce clusters that are themselves polysemous, which may not be a desirable property of a word sense induction algorithm. As a solution to this problem, Tomuro et al. suggest using *feature domain similarity*, which refers to the similarity between the *features* of items rather than the similarity between the items themselves. The domain feature similarity score is incorporated in a modified version of the clustering by committee algorithm, in which the algorithm is run twice, using the output of the first run as input to the second run. The idea is that this iterative approach may enable the algorithm to utilize higher-order features, and that this will inhibit the formation of polysemous clusters, since the domain feature similarity of a polysemous cluster will be lower than the score for a monosemous cluster.

(Koptjevskaja Tamm and Sahlgren, 2014) also leverage on the idea of using feature similarity as the basis of sense clustering. The approach, called *syntagmatically labeled partitioning*, relies on a distributional semantic model that encodes sequential as well as substitutable relations. The method essentially sorts the k nearest (substitutable) neighbors according to which sequential connections they share. The resulting partitioning of the near-

est distributional neighbors does not only constitute a word-sense induction, but it also provides *labels* for the induced senses in the form of the sequential connections the neighbors share.

4 Neighborhood Graphs

Many of the previous approaches to word-sense induction mentioned in the previous section operate at a global level, utilizing global structural properties of the semantic spaces, e.g. by matrix factorization techniques. We believe this is as ill-advised as setting a global k or radius for the nearest neighbor search, since it is the *local* structures that are important when analyzing nearest neighbors. Other approaches to word-sense induction use various forms of clustering techniques. However, previous studies of the intrinsic dimensionality of distributional semantic spaces using fractal dimensions indicate that neighborhoods in semantic space have a *filamentary* rather than clustered structure (Karlgrén et al., 2008).

We therefore propose the use of *topological* models that take the *local* structure of neighborhoods in semantic space into account. The method proposed here performs no global clustering, does not concern itself with grammatical preprocessing or parsing, and the distributional vectors are taken as is. The approach discovers different word senses from the local structure of neighborhoods, given nothing but similarities between points. As such it is easy to test on widely different vector models, as long as there exists a well behaved similarity function. The proposed approach not only answers the question which other terms are similar to a given term, but also *how* are they similar.

Relative neighborhoods are examples of *empty region graphs* (Cardinal et al., 2009), where points are neighbors if some region between them is empty. For relative neighborhood graphs the region between two points a and c belonging to some set of points V is defined as the intersection of the two spheres with centers in a and c , with radius $d(a, c)$. In other words, a point b lies between points a and c if it is closer to both a and c than a and c are to each other, and if no such point b exists, a and c are neighbors. Illustrations of this can be seen in Figure 4.

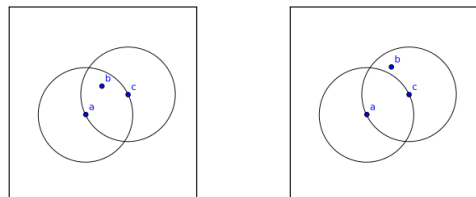


Figure 4: Example of when point b is between point a and c (left), and when it is not (right).

Such neighborhoods have been argued to better preserve local topology (Bremer et al.,), and be more robust to deformations of the data than k -NN neighborhoods (Correa and Lindstrom, 2012) as they in some sense contain information about direction whereas k -NN neighborhoods only contain information about distance. Going back to the "suit" example, we can see that if "suit" in the law sense is more similar to the composite "suit" than to its clothes sense, and vice versa, then the composite v_{suit} lies between $v_{suit|law}$ and $v_{suit|clothes}$. This in turn means that out of those two points, both are relative neighbors to "suit", and neither of them lies between the other and "suit".

Formally, the set of points between two points $a, c \in V$ can be characterized and computed in the following way:

$$\text{between}(V, a, c) = \{b | b \in V, b \text{ lies between } a \text{ and } c\}$$

$$\text{rng-nbh}(V, a) = \{c | c \in V, \text{between}(V, a, c) = \emptyset\}$$

$$E_{\text{rng}}(V) = \{(a, b) | a \in V, b \in \text{rng-nbh}(V, a)\}$$

where E_{rng} is the undirected edge set of the RNG. The function $\text{between}(V, a, c)$ can be straightforwardly translated to an algorithm taking $\mathcal{O}(|V|)$ time, making the rng-nbh function take $\mathcal{O}(|V|^2)$ time, which in turn makes the computation of the complete graph take $\mathcal{O}(|V|^3)$ time.⁵ Clearly unfeasible, but we have not found any alternatives that performs better in the high dimensional case.⁶

In (Correa and Lindstrom, 2012) it is noted that the intersection of the relative neighborhood graph and the k -NN graph is a more feasible alternative:

$$k\text{-rng-nbh}(V, a) = \text{rng-nbh}(V', a)$$

where $V' = k$ nearest neighbors of a

Given a precompiled — i.e. constant time — k -NN lookup, the above takes $\mathcal{O}(k^2)$ time, so using a heap-based $\mathcal{O}(|V| \lg k)$ k -NN algorithm results in an algorithm taking $\mathcal{O}(k^2 + |V| \lg k)$ time.

The same idea can be used to build a tree structure — here called *relative neighborhood tree* — rooted in a reference word a , in the following way:

$$\text{rbnh-tree}(V, a) = \{(c, \arg \min_{b \in B_c} d(b, c)) | c \in V\}$$

where $B_c = \{a\} \cup \text{between}(V, a, c)$

This can easily be restricted to the k -nearest neighbors of a in much the same way as above.

Computing this for a point a produces a tree where the direct children of a are its relative neighbors, and the parent of a point c further down the tree is the point between a and c that is closest to

⁵ Assuming a constant time distance function.

⁶ It should be noted that there are more efficient algorithms for lower dimensional situations.

c. Figure 5 illustrates what the resulting structure looks like for "heart" on its 100 nearest neighbors in the PMI model. Note that the root "heart" (at the mid-left in the graph) only has two relative neighbors: "cardiac" and "soul." One advantage of using this type of structure for the neighborhood is that it enables us to examine various depths of the tree. Depth one includes only the direct neighbors ("cardiac" and "soul"); depth two includes all neighbors two steps away in the graph: "disease," "coronary," "pulmonary," "cardiovascular," "ventricular," and "failure," which are all children to "cardiac;" depth three also includes the neighbors "kidney," "severe," "complications," and "diseases" as children to "disease," "atrial" and "arrhythmias" as children to "ventricular," and the neighbors "respiratory," "lung," "tumors," "aortic" as children to "pulmonary." This tree structure can be used to identify neighbors that are themselves polysemous (c.f. the critique mentioned in Section 3 of clustering-based approaches to word-sense induction that they may produce polysemous clusters (Tomuro et al., 2007)). One example is the neighbor "disease" at depth two, which has six children that refer to different aspects of disease.

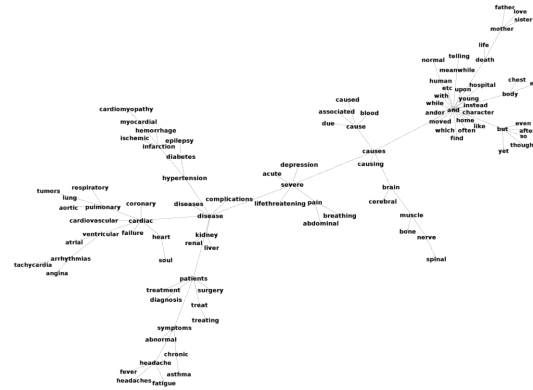


Figure 5: Relative neighborhood tree for "heart" in the PMI model, restricted to the 100 closest neighbors.

This tree-structure is thus quite useful in the context of word-sense induction, since the branching structure indicates different usages, and the depth factor enables us to calibrate the granularity of the induced word senses. If we only consider direct neighbors (i.e. depth one), and set $k = V$ (i.e. we do an exhaustive nearest neighbor search), we will extract all terms that have a direct connection to the reference term. We refer to this neighborhood as the *semantic horizon*. At the most coarse level of analysis, this is the neighborhood that represents the main induced senses of a term. Tables 2 and 3 provide examples of 1000-RNG neighborhoods.

These examples demonstrate some interesting

Table 2: Relative neighborhood of the words "suit," "orange," and "heart" in three different semantic models. The numbers in parenthesis indicate the k -NN ranks of the neighbors.

PMI	GloVe	skipgram
suit		
suits (1) dress (2) lawsuit (10) dinosaur (53) costly (60) option (76) counterparts (99) predator (107) trump (109) : :	suits (1) lawsuit (2) mobile (33) gundam (34) trump (55) zoot (133) rebid (423) serenaders (458) hev (987)	suits (1) lawsuit (2)
orange		
yellow (1) lemon (16) : :	yellow (1) ktype (12) lemon (14) citrus (17) jersey (21) cherry (24) county (26) peel (42) jumpsuits (57) : :	redorange (1)
heart		
cardiac (1) soul (22) hearts (183) ashtray(641) rags(771)	my (1) blood (2) throbs (3) suffering (4) brain (6) cardiac (8) hearts (11) throb (17) lungs (22) : :	congestive (1) hearts (2)

similarities and differences between the three models. First of all, there are some direct neighbors that are present in all three models: "suit" has "suits" and "lawsuit" as direct neighbors in all three models, "heart" has "hearts," "service" has "services," and "above" has "below". Plural forms are of course reasonable neighbors of their singular counterparts in a semantic model, but their usefulness for word-sense induction can perhaps be questioned. Taking "suits" to indicate the *garment*-related sense of "suit," all three models produce both a *garment*-related and *law*-related sense. For "orange," the skipgram model only represents the *color* sense, while the PMI and GloVe models also feature a *fruit* sense. For "heart," all three models have a *disease* sense (represented by the neighbors "cardiac" in the PMI and GloVe models, and the neighbor "congestive" in the skipgram model), and an *organ* sense (represented by the plural form "hearts"). "Service" is a comparably vague term that has a number of different senses in the PMI and GloVe models, but only one in the skipgram model. "Bad" produces both a *negativity* sense and a *German spa town*-sense in all three models, both only the GloVe and skipgram models have a separate antonym sense ("good" is not a direct

neighbor in the PMI model). "Above" has both the antonym and direct neighbors relating to measurements in all three models.

GloVe produces the most branched neighborhoods, with a large number of direct neighbors, while the skipgram model produces the least branched neighborhoods with at most a couple of direct neighbors for each term. The PMI model is somewhere in between. One reason why GloVe produces such branching neighborhoods is that GloVe seems to capture not only semantic relations but also a significant amount of sequential relations. Many of the neighbors in the k -RNG for GloVe come from collocations: the k -RNG for "suit" includes "mobile" and "gundam," which come from the collocation "mobile suit gundam" that is an anime series, "trump" that relates to "trump suits" in card games, "serenaders" that refer to the retro string band "cheap suit serenaders," and the very distant neighbor "hev," which comes from "hev suit" that relates to the Half-life series of first person shooter games. For "orange" we find "ktype" that comes from the collocation "ktype stars," which is another term for "orange dwarfs", as well as the collocations "orange peel", "orange county", "orange jumpsuit", "cherry orange", and so on. The k -RNG for "heart," "service," "bad," and "above" also feature a number of collocations for the GloVe model. There are also some examples of neighbors from sequential relations in the PMI model (e.g. "costly" as neighbor to "suit" from the collocation "costly suit," "luck" and "donnersbergkreis" as neighbors to "bad" from the collocations "bad luck" and "bad donnersbergkreis"), but this tendency is not at all as pronounced as it is for the GloVe model.

The PMI and GloVe models produce the structurally most similar RNGs, with on average a handful of direct neighbors, of which some can be very distant. The skipgram model on the other hand produces very few direct neighbors. This led us to look further into the structural properties of neighborhoods in the skipgram model. An interesting observation — and possible complication — is that the neighborhoods in the skipgram model are highly asymmetric: the first neighbor of "information" is "informations", whereas "information" is only the 1829th neighbor of "informations." While such asymmetry occurs in all models, it seems much more prevalent in the skipgram model. Figure 6 demonstrates this: each point corresponds to a random word pair (a, b) with x corresponding to where b is in the ordered list of a 's neighbor, and y to where a is in the ordered list of b 's neighbors, or equivalently: x is the number of points within $d(a, b)$ of a and y is the number of points within $d(a, b)$ of b . This implies that the local densities vary much more in the skipgram

Table 3: Relative neighbors to the word "service," "bad," and "above" in three different semantic models. The numbers in parenthesis indicate the k -NN ranks of the neighbors.

PMI	GloVe	skipgram
service		
services (1) network (2) operates (8) launched (18) served (22) intercity(34)	services (1) operated (3) serving (6) military (17) duty (20) passenger (21) dialaride (644) aftersales (759) limitedstop (802)	services (1)
bad		
terrible (1) that (2) luck (39) unfortunate (70) stalling (276) donnersbergkreis (860) rancid (980)	good (1) kissingen (2) ugly (45) nasty (48) dirty (106) omen (328) conkers (360) karma (952)	nauheim (1) good (2) dreadful (5)
above		
below (1) around (2) feet (5) measuring (29) beneath (36) columns (62) atop (102)	below (1) level (2) height (3) just (4) stands (10) lower (11) beneath (12) rise (21) sea (30) : :	below (1) 500ft (2)

model than in the others, which can complicate the choice of k in the k -RNG algorithm.

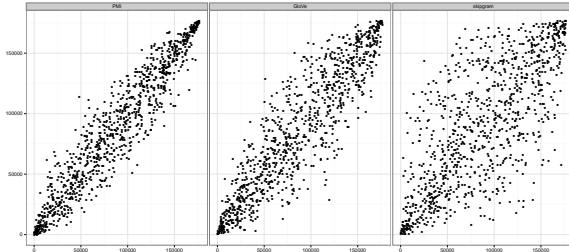


Figure 6: neighborhood reciprocity in the different models.

5 Conclusions

In this paper we have discussed the question how to query semantic models, which is a question that has been long neglected in research on computational semantics. Nearest neighbor search (or k -NN) is often treated as the only available option, which leads to misunderstandings regarding how semantic models represent and handle vagueness and polysemy. We have argued that the structure — or topology — of the local neighborhoods in semantic models carry useful semantic information regarding the different usages — or senses — of a term, and that such topological properties thus can be used to analyze polysemy and to perform

word-sense induction. We have also argued that the topology of the local neighborhoods in semantic models can be used for selecting a relevant set of neighbors — a factor we have referred to as the semantic horizon.

We have introduced relative neighborhood graphs (RNG) as an alternative to standard k -NN, and we have illustrated how k -RNG can be used as a tool for analyzing the topology of local neighborhoods in semantic models. We have exemplified relative neighborhoods in three different well-known semantic models; the standard PMI model, as well as the more recent GloVe and skipgram models. The examples provided in this paper demonstrate that k -RNG can be used for word-sense induction, but that such topological methods are more suitable to use for certain types of semantic models. The k -RNG for the PMI and GloVe models produce pleasant results, while the skipgram model, with its big local variations in density, produces less informative results. It's quite possible that the complete RNG overcomes these problems, but that does not seem a feasible solution.

This illustrates how k -RNG can be used as a tool to gain insight into the topological properties of different models. We have also observed that the GloVe model often produces neighbors that correspond to various collocations, which means that this model is not strictly a *semantic* representation, since it confounds substitutable and sequential relations. A more sophisticated tokenization, taking n -grams into account, might alleviate this. The standard PMI model is nowadays often overlooked in favor of more recent neural network-inspired models, but our results indicate that the PMI model has a number of comparatively attractive properties that are useful for linguistic applications such as word-sense induction.

References

- [Agirre et al.2006] Eneko Agirre, David Martínez, Oier López de Lacalle, and Aitor Soroa. 2006. Two graph-based algorithms for state-of-the-art wsd. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, EMNLP '06*, pages 585–593, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Arya et al.1998] Sunil Arya, David M. Mount, Nathan S. Netanyahu, Ruth Silverman, and Angela Y. Wu. 1998. An optimal algorithm for approximate nearest neighbor searching fixed dimensions. *J. ACM*, 45(6):891–923, November.
- [Baroni et al.2014] Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count,

- predict! a systematic comparison of context-counting vs. context-predicting semantic vectors. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, volume 1.
- [Bentley1975] Jon Louis Bentley. 1975. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, September.
- [Biemann2006] Chris Biemann. 2006. Chinese whispers: An efficient graph clustering algorithm and its application to natural language processing problems. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, TextGraphs-1, pages 73–80, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Bremer et al.] Peer-Timo Bremer, Ingrid Hotz, Valerio Pascucci, and Ronald Peikert. Topological methods in data analysis and visualization iii.
- [Brin and Page1998] Sergey Brin and Larry Page. 1998. The anatomy of a large-scale hypertextual web search engine. In *Seventh International World-Wide Web Conference (WWW 1998)*, pages 107–117. Elsevier Science Publishers B. V., April.
- [Brody and Lapata2009] Samuel Brody and Mirella Lapata. 2009. Bayesian word sense induction. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 103–111, Athens, Greece, March. Association for Computational Linguistics.
- [Cardinal et al.2009] Jean Cardinal, Sébastien Collette, and Stefan Langerman. 2009. Empty region graphs. *Computational geometry*, 42(3):183–195.
- [Correa and Lindstrom2012] Carlos D Correa and Peter Lindstrom. 2012. Locally-scaled spectral clustering using empty region graphs. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1330–1338. ACM.
- [Dinu and Lapata2010] Georgiana Dinu and Mirella Lapata. 2010. Measuring distributional similarity in context. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP ’10, pages 1162–1172, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Dorow and Widdows2003] Beate Dorow and Dominic Widdows. 2003. Discovering corpus-specific word senses. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 2*, pages 79–82. Association for Computational Linguistics.
- [Erk and Padó2010] Katrin Erk and Sebastian Padó. 2010. Exemplar-based models for word meaning in context. In *Proceedings of the ACL 2010 Conference Short Papers*, ACLShort ’10, pages 92–97, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Indyk and Motwani1998] Piotr Indyk and Rajeev Motwani. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613, New York, NY, USA. ACM.
- [Jurgens and Stevens2010] David Jurgens and Keith Stevens. 2010. Hermit: Flexible clustering for the semeval-2 wsi task. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 359–362, Uppsala, Sweden, July. Association for Computational Linguistics.
- [Karlgrén et al.2008] Jussi Karlgrén, Anders Holst, and Magnus Sahlgrén. 2008. Filaments of meaning in word space. In Craig Macdonald, Iadh Ounis, Vassilis Plachouras, Ian Ruthven, and Ryan W. White, editors, *ECIR*, volume 4956 of *Lecture Notes in Computer Science*, pages 531–538. Springer.
- [Klapaftis and Manandhar2008] Ioannis P. Klapaftis and Suresh Manandhar. 2008. Word sense induction using graphs of collocations. In *Proceedings of the 2008 Conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, pages 298–302, Amsterdam, The Netherlands, The Netherlands. IOS Press.
- [Koptjevskaja Tamm and Sahlgrén2014] Maria Koptjevskaja Tamm and Magnus Sahlgrén. 2014. Temperature in word space: Sense exploration of temperature expressions using word-space modelling. In Benedikt Szmrecsanyi and Bernhard Wälchli, editors, *Aggregating dialectology, typology, and register analysis*, pages 231–267. De Gruyter.
- [Lau et al.2012] Jey Han Lau, Paul Cook, Diana McCarthy, David Newman, and Timothy Baldwin. 2012. Word sense induction for novel sense detection. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 591–601. Association for Computational Linguistics.
- [Levy and Goldberg2014] Omer Levy and Yoav Goldberg. 2014. Neural word embedding as implicit matrix factorization. In *Advances in Neural Information Processing Systems*, pages 2177–2185.
- [Marco and Navigli2013] Antonio Di Marco and Roberto Navigli. 2013. Clustering and diversifying web search results with graph-based word sense induction. *Computational Linguistics*, 39(3):709–754.

- [Mikolov et al.2013] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26 (NIPS'13)*, pages 3111–3119.
- [Pantel and Lin2002] Patrick Pantel and Dekang Lin. 2002. Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '02, pages 613–619, New York, NY, USA. ACM.
- [Pedersen2010] Ted Pedersen. 2010. Duluth-wsi: Senseclusters applied to the sense induction task of semeval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation, SemEval '10*, pages 363–366, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Pennington et al.2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*.
- [Pereira et al.1993] Fernando Pereira, Naftali Tishby, and Lillian Lee. 1993. Distributional clustering of english words. In *Proceedings of the 31st annual meeting on Association for Computational Linguistics*, ACL '93, pages 183–190, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Purandare and Pedersen2004] Amruta Purandare and Ted Pedersen. 2004. Word sense discrimination by clustering contexts in vector and similarity spaces. In *HLT-NAACL 2004 Workshop: Eighth Conference on Computational Natural Language Learning (CoNLL-2004)*, pages 41–48, Boston, Massachusetts, USA, May. Association for Computational Linguistics.
- [Reisinger and Mooney2010] Joseph Reisinger and Raymond J. Mooney. 2010. Multi-prototype vector-space models of word meaning. In *Proceedings of the 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-2010)*, pages 109–117.
- [Schütze1998] Hinrich Schütze. 1998. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, march.
- [Séaghdha and Korhonen2011] Diarmuid Ó Séaghdha and Anna Korhonen. 2011. Probabilistic models of similarity in syntactic context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 1047–1057, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Tomuro et al.2007] Noriko Tomuro, Steven L. Lytinen, Kyoko Kanzaki, and Hitoshi Isahara. 2007. Clustering using feature domain similarity to discover word senses for adjectives. In *Proceedings of the First IEEE International Conference on Semantic Computing (ICSC 2007), September 17-19, 2007, Irvine, California, USA*, pages 370–377.
- [Van de Cruys and Apidianaki2011] Tim Van de Cruys and Marianna Apidianaki. 2011. Latent semantic word sense induction and disambiguation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1476–1485, Portland, Oregon, USA, June. Association for Computational Linguistics.
- [van Dongen2000] Stijn van Dongen. 2000. *Graph Clustering by Flow Simulation*. Ph.D. thesis, University of Utrecht.
- [Velldal2005] Erik Velldal. 2005. A fuzzy clustering approach to word sense discrimination. In *Proceedings of the 7th International conference on Terminology and Knowledge Engineering*, Copenhagen, Denmark.
- [Véronis2004] Jean Véronis. 2004. Hyperlex: lexical cartography for information retrieval. *Computer Speech & Language*, 18(3):223–252.
- [Yao and Van Durme2011] Xuchen Yao and Benjamin Van Durme. 2011. Nonparametric bayesian word sense induction. In *Proceedings of TextGraphs-6: Graph-based Methods for Natural Language Processing*, TextGraphs-6, pages 10–14, Stroudsburg, PA, USA. Association for Computational Linguistics.